UNIVERSITEIT VAN AMSTERDAM

DEEP LEARNING

PRACTICAL 2

RECURRENT NEURAL NETWORKS AND GRAPH NEURAL NETWORKS

Andreea Teodora Patra
Student Number:13365169
andreea.patra@student.uva.nl

December 1, 2020

1 Vanilla RNN

Question 1.1 The equations derived can be seen in the image

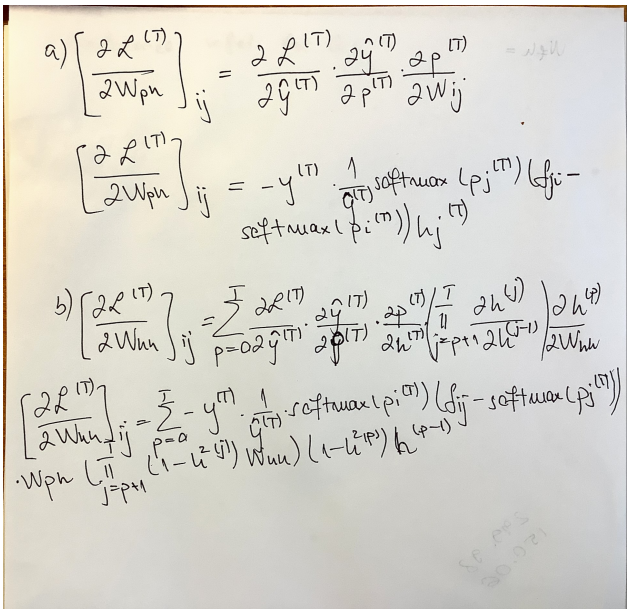

Figure 1. Part a and b

c) We can see the temporal difference on the second gradient as it depends on the gradients up until timestep T which can be seen in the sum factor.

Question 1.2 a)The gate i tries to understand how important is the input for the prediction and it uses a sigmoid nonlinearity. The gate f tries to understand how important is the past state and it uses a sigmoid nonlinearity. The gate g tries to understand which potential memory to use to update the state and it uses a tanh nonlinearity. Using the sigmoid function which has an output between 0 and 1, we choose how much of the input we let to go through the network.

1.1 LSTMs in Pytorch

Question 1.3 For this part, we implement LSTM for the binary palindrome sequence. The accuracy and the loss curves can be seen in the following figures.

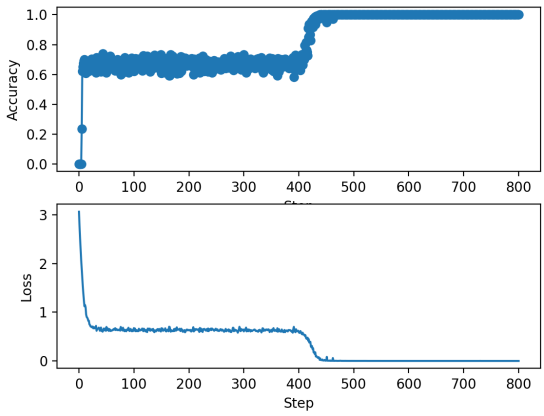

Figure 2. Accuracy and loss curves for T=10

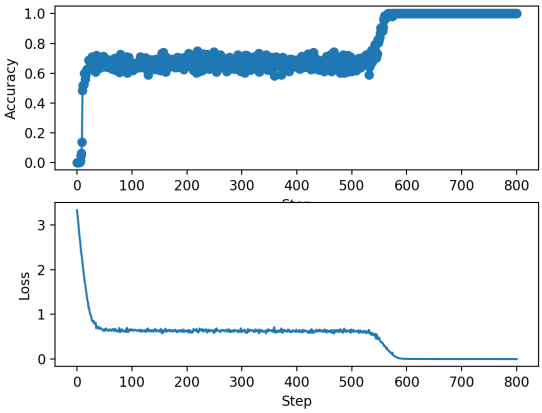

Figure 3. Accuracy and loss curves for T=20

Both networks have converged to perfect accuracy at around the 800 step and as a result early stopping was implemented. Then the accuracy of the network is averaged over 3 different seeds and we get an accuracy of 1.0 and a standard deviation of 0.0. The model starts with 0 accuracy as it has not had time to learn the dependencies between the numbers and then rapidly jumps to 0.6. When the sequence length is bigger, the model converges slower. We can also see from the graphs that for T=20, there are more oscillations in the accuracy compared to T=10. We can conclude that it may be harder to record all the dependencies on longer sequences.

Question 1.4 For this part, we use the same dataset, but with LSTM with peephole connections. Both networks have converged to perfect accuracy at around the 800 step and as a result early stopping was implemented. Then the accuracy of the network is averaged over 3 different seeds and we get an accuracy of 1.0 and a standard deviation of 0.0. We can see that the initial loss for this method is lower compared to simple LSTM. The accuracy and loss curves can be seen in the following graphs:

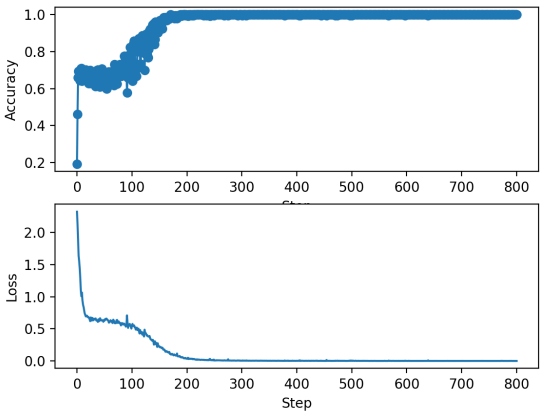

Figure 4. Accuracy and loss curves for T=10

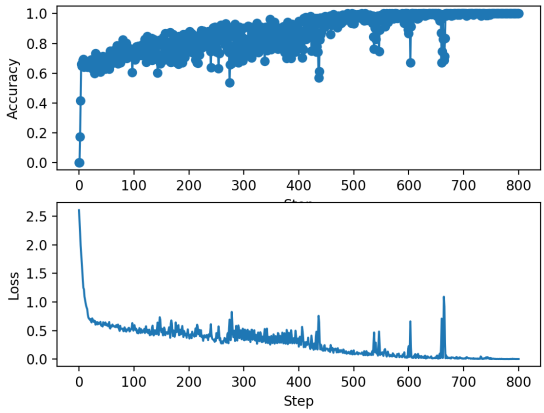

Figure 5. Accuracy and loss curves for T=20

2 Recurrent Nets as Generative Models

Question 2.1a) In this question we use the recurrent network as sequence to sequence mapping. One character at a time is fed into the recurrent network and it tries to predict the next character from the sequence. The teacher forcing is ensured by having the targets as the sequence shifted by one. In this way, the input to the next token is not the predicted value from the network, it is the true value. In Figure 1 , you can see the accuracy and the loss curves during training.

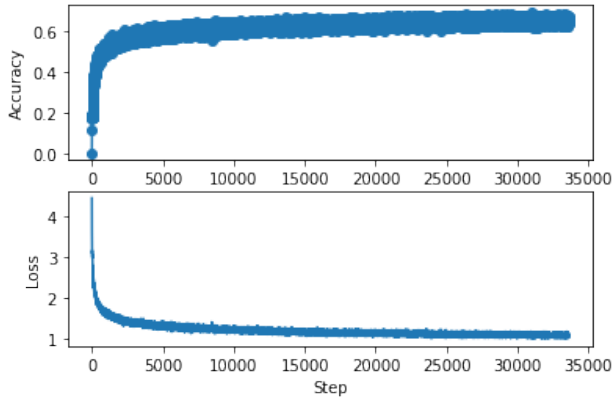

Figure 6. Accuracy and loss curves

When training we have used 336000 steps and upon testing we have observed that the accuracy and loss started not having big improvements and it was stabilised. The learning rate used is 0.002 and the optimiser is Adam. We have chosen Adam as it has performed good in the classification of number sequences as well. The learning rate is chosen quite small as we do not have a lot of data and we are working with small sequences. We also add clipping of the gradient so that we avoid explosions and make smaller steps towards the minimum. The batch size is kept at 64, the hidden layer is 128 and the embedding dimension is 87. The accuracy oscillates around 0.66 in the end.

Question 2.1b) For this part we generate sequences of different lengths(30,40,60) using random sampling of the next token. We generate 5 sample text for the 1/3, 2/3 and 3/3 step of the training iteration.
For length 30 we have the following sentences for 1/3 iteration:

- 91. LIMITED THOS AGITER THIS
- ith the world was so that the
- the second stream. Then the se

- .E.3. Except the tree and sai
- en the second stream. Then the

For length 30 we have the following sentences for 2/3 iteration:

- hey were so seemed to see if a
- ou must go away, and the secon
- ut of the water, and the cook
- the cat was the straw that th
- 'I will not go to the water.'

For length 30 we have the following sentences for 3/3 iteration:

- 18559), were so that the wolf
- ' said the fisherman was so th
- MINDAED AND THE SEVEN LITTLE
- the castle was allowed to the
- THE FOUNDATION, THE BRE

We can see that the network usually outputs with a high probability the blank space. As well we can see a lot of words that are unfinished and the network still has selected. Over the training process, we can see that the sentences make more sence and they seem to have must structure and more characters as well and not so many blank spaces. The remaining sentences can be see in the 'summaries.txt' file that is uploaded with this project. Looking at the longer sequences, we can still see that we can comprehensible sentences even if we are only training on 30 tokens. We can also see that the quotation marks have yielded a higher probability which could be expected if the text has more dialogue. We do not have many commas or other punctuation marks except the quotation.

Question 2.1c) The text generations can be found in the file 'summaries.txt' Temperature helps with the exploration of the vocabulary. Instead of always selecting the token with the maximum probability, we choose to reweigh it. By doing so, we can get more creative and more diversified text generations. The temperature of 2 has resulted sentences that are not coherent. This could be due to the small number of tokens used in training. As expected, the greedy sampling has more repetition of words than the one with temperature.

3 Graph Neural Networks

3.1 GCN Forward Layer

Question 3.1a) From equation 44 and the lecture notes, we understand that we need to multiply with the adjacency matrix so that we recover the features in the neighbourhood, therefore the nodes that one node is interacting with. Moreover, we multiply with W so that we convolve that neighbourhood, the same way we would do it in CNNs. The structural information is preserved by multiplying with the adjacency matrix. As well, by having knowledge of the connections between the nodes, we can understand how the features flows through the graph.

Question 3.1b) If we think about spectral graph convolution network, we can see that the parameter complexity is one of the drawbacks. They require to see the whole graph to be processed simultaneously, but this could lead to complications for millions of nodes. One solution would be to use the spatial based approach which aggregates the information from neighbouring nodes.

Question 3.2a) The adjacency matrix for the graph is as follows: $\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$

Question 3.2b)

4 Graph attention networks

Question 3.3 So that we can perform attention over the neighbouring nodes of i we need to add an attention weight. The final equation would be $h_i^{l+1} = \sigma\Big(\sum_{j \in N(i)} \alpha_{ij} W^l h_j^l\Big)$. Attention is that so that we would not weigh every neighbour equally and more depending on their actual values.

Question 3.4 One real world application is in the chemistry domain. GCNs can be used to learn the embeddings of a molecule structure and can then be trained to generate new molecules. As well, in this paper [1], GCNs are used to predict the chemical reactivity. Another domain where it could be applied is social analysis. We can easily represent all of our interactions on social media platforms onto a graph. This could then be used to predict multiple features of each individual based on its interaction with their neighbours. As well, targeted advertising campaigns can be derived from this graph.

5 Comparing and Combining GNNs and RNNs

Question 3.5a) If we are talking about temporal structured data, RNNs can outperform GNNs. We cannot assign an order on the nodes, thus the temporal dependence would be lost. Therefore, RNNs are more suitable for dynamical data, while as GNNs are used more in structured data, such as the chemical reactions.

Question 3.5b) One application where RNNs and GNNs can be used is in videos. We would need GNNs so that we can fit the structure of the actions across moving frames and across one frame. But we will also need RNNs so that the temporal dimension is not lost.

6 References

[1] A graph-convolutional neural network model for the prediction of chemical reactivity, Chem. Sci., 2019,10, 370-377